

Fig. 1

$$\begin{aligned}
 629 &= 2^9 \quad 2^8 \quad 2^7 \quad +2^6 \quad +2^5 \quad +2^4 \quad \quad +2^2 \quad \quad +2^0 \\
 &= \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline .1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ \hline +1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ \hline \end{array} \\
 &= 2^9 + (2^8 - 2^7 - 2^6) + 2^5 + 2^4 + (2^3 - 2^2) + (2^1 - 2^0) \\
 \\
 628 &= 2^9 \quad \quad \quad +2^6 \quad +2^5 \quad +2^4 \quad \quad +2^2 \\
 &= \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ \hline +1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ \hline \end{array} \\
 &= \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline & & & & & & & & -1 & \\ \hline \end{array} \\
 &= 2^9 + (2^8 - 2^7 - 2^6) + 2^5 + 2^4 + (2^3 - 2^2) + (2^1 - 2^0)
 \end{aligned}$$

Fig. 2

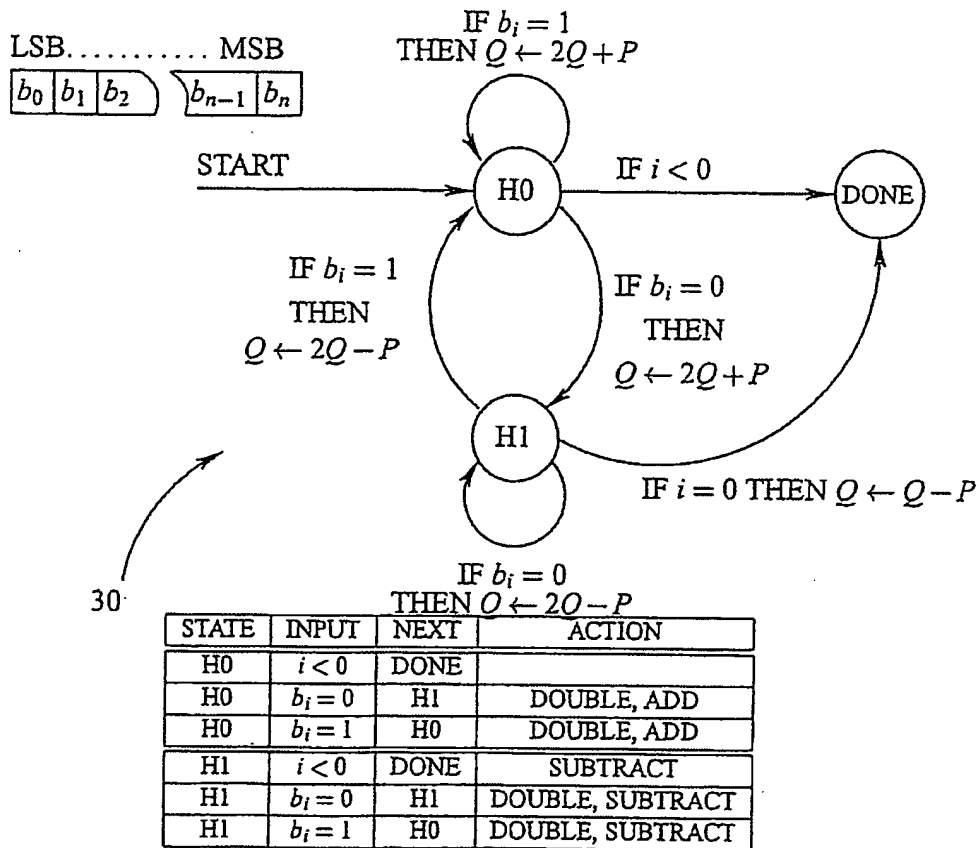


Fig. 3



```

BEGIN:
  i := N          ; START FROM MSB          L1
  Q := 0          ; INITIALIZE ACCUMULATOR  L2
  H := 0          ; INITIALIZE STATE        L3

LOOP:             ; FOR ALL BITS
  Q := Q + Q      ; DOUBLE ACCUMULATOR      L4
  IF H = 0        ; IF H STATE IS SET        L5
    Q := Q + P    ; ADD BASE POINT TO ACCUMULATOR L6
    GOTO ENDLOOP ;                          L7
  ELSE
    Q := Q - P    ; SUBTRACT BASE POINT      L8
    GOTO ENDLOOP ;                          L9

ENDLOOP:
  H :=  $\overline{b[i]}$  ; SET H STATE TO COMPLEMENT OF b[i]  L10
  i := i - 1      ; PROCESS NEXT BIT          L11
  IF i ≥ 0        ; IF BIT EXISTS             L12
    GOTO LOOP     ; CONTINUE AT TOP OF LOOP    L13
  IF H = 0        ; IF EXITING FROM H = 0 STATE L14
    Q := Q + (-P) ; CORRECT RESULT BY FINAL SUBTRACT L15
  END             ;                          L16

```

Fig. 4



```

BEGIN:
    i := N          ; START FROM MSB          LL1
    Q := 0          ; INITIALIZE ACCUMULATOR  LL2

H0:      ; STATE ENTRY POINT
    Q := Q + Q      ; DOUBLE ACCUMULATOR      LL3
    Q := Q + P      ; ADD BASE POINT TO ACCUMULATOR  LL4
    GOTO ENDLOOP    ; BRANCH TO END OF LOOP TESTS  LL5

H1:      ; STATE ENTRY POINT
    Q := Q + Q      ; DOUBLE ACCUMULATOR      LL6
    Q := Q + (-P)   ; SUBTRACT BASE POINT FROM ACCUMULATOR  LL7
    GOTO ENDLOOP    ; BRANCH TO END OF LOOP TESTS  LL8

ENDLOOP: ; END OF LOOP TESTS
    IF b[i] = 1     ; IF CURRENT BIT IS SET      LL9
        GOTO NEXT H0 ; FOLLOW H0 PATH          LL10
    ; ELSE FALL INTO H1 PATH

NEXT H1: ; H1 PATH
    i := i - 1      ; PROCESS NEXT BIT          LL11
    IF i > 0         ; IF BIT EXISTS            LL12
        GOTO H1     ; EXECUTE H1 STATE          LL13
    Q := Q + (-P)   ; ELSE CORRECT RESULT AND END  LL14
    END             ; ELSE END                  LL15

NEXT H0: ; H0 PATH
    i := i - 1      ; PROCESS NEXT BIT          LL16
    IF i > 0         ; IF BIT EXISTS            LL17
        GOTO H0     ; EXECUTE H0 STATE          LL18
    END             ; ELSE END                  LL15

```

Fig. 5



```

BEGIN:
   $i := N$ 
   $Q := 1$ 

H0:
   $Q := Q \cdot Q (Q^2)$ 
   $Q := Q \cdot M$ 
  GOTO ENDLOOP

H1:
   $Q := Q \cdot Q$ 
   $Q := Q/M (Q \cdot M^{-1})$ 

60 ENDLOOP:
  IF  $b[i] = 1$  GOTO ENDLOOP

NEXT H1:
   $i := i - 1$ 
  IF  $i > 0$ 
    GOTO H1
   $Q := Q/M$ 
  END

NEXT H0:
   $i := i - 1$ 
  IF  $i > 0$ 
    GOTO H0
  END
  
```

Fig. 6

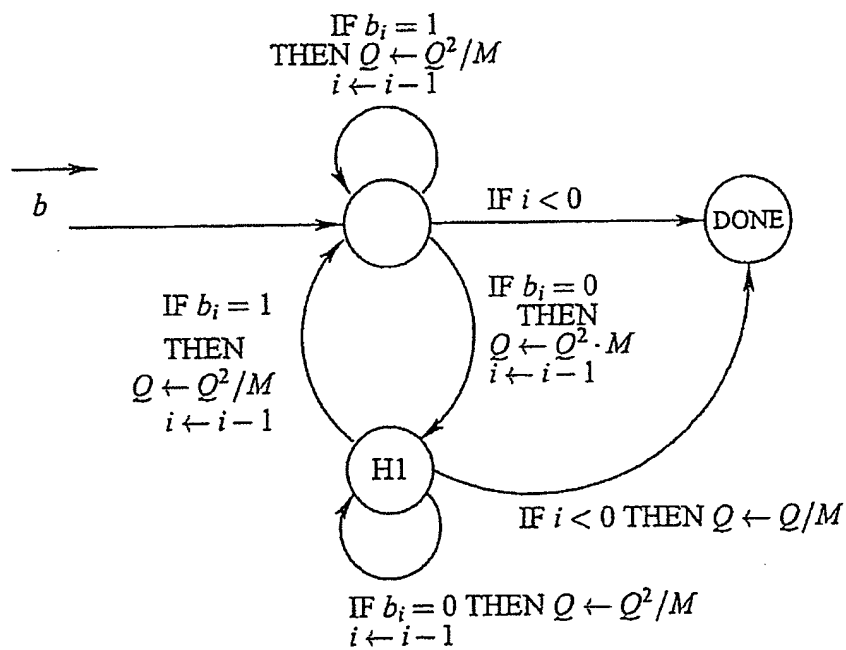
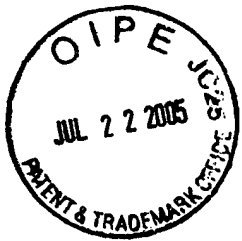


Fig. 7

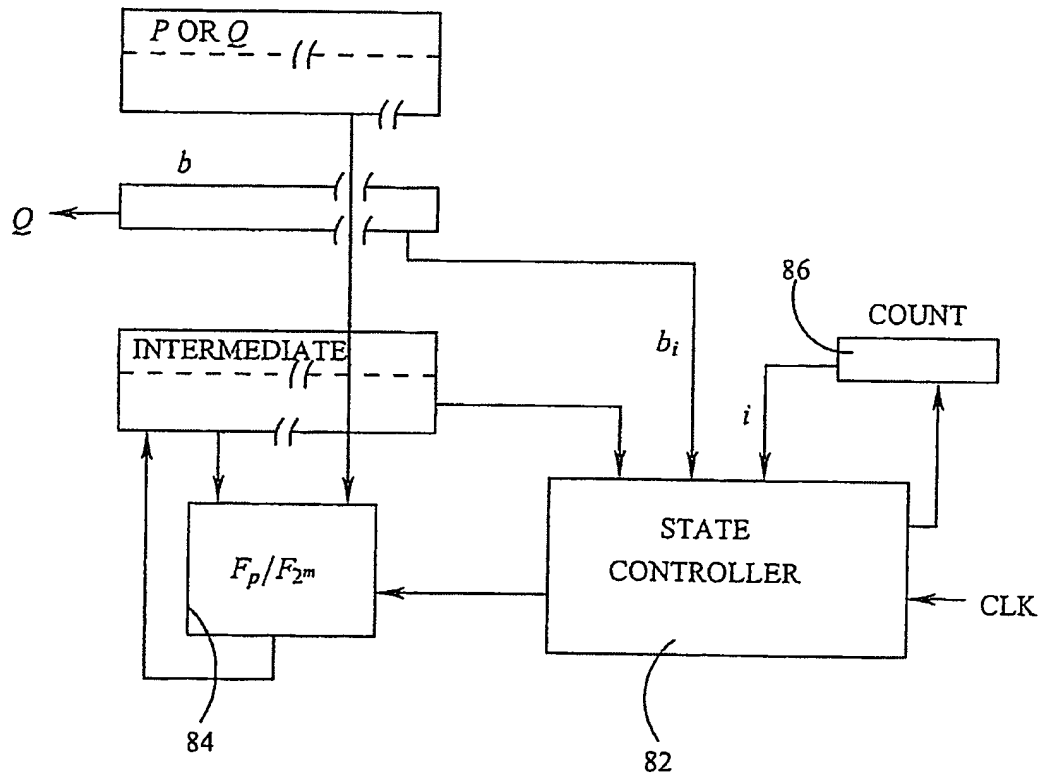
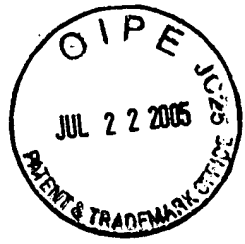


Fig. 8